**APPENDIX**

-- Select the entire dataset for review

USE hap780project;

SELECT \* FROM dbo.drugs\_dataset;

---- DATA PREPROCESSING ----

-- Change data types for each column to ensure consistent data formats

SELECT

TRY\_CAST (id as int) as ID, -- Ensure ID is an integer

TRY\_CAST (drugName as nvarchar(max)) as DrugName, -- Ensure drugName is text

TRY\_CAST(condition as nvarchar(max)) as DiseaseCondition, -- Ensure condition is text

TRY\_CAST (review as nvarchar(max)) as Review\_of\_Drug, -- Ensure review is text

TRY\_CAST (rating as int) as Rating, -- Ensure rating is an integer

TRY\_CAST (date as date) as Date, -- Ensure date is in date format

TRY\_CAST (usefulCount as int) as UsefulDrugCount -- Ensure usefulCount is an integer

FROM dbo.drugs\_dataset;

-- Identifying null values in each critical column

SELECT \* FROM dbo.drugs\_dataset WHERE ID IS NULL; -- Check for null IDs

SELECT \* FROM dbo.drugs\_dataset WHERE drugName IS NULL; -- Check for null drug names

SELECT \* FROM dbo.drugs\_dataset WHERE condition IS NULL; -- Check for null conditions (1194 found)

SELECT \* FROM dbo.drugs\_dataset WHERE review IS NULL; -- Check for null reviews

SELECT \* FROM dbo.drugs\_dataset WHERE Rating IS NULL; -- Check for null ratings

SELECT \* FROM dbo.drugs\_dataset WHERE Date IS NULL; -- Check for null dates

SELECT \* FROM dbo.drugs\_dataset WHERE usefulCount IS NULL; -- Check for null usefulCount

-- Remove rows where the 'condition' column is null

DELETE FROM dbo.drugs\_dataset WHERE condition IS NULL; -- 1194 rows removed

-- Trim whitespace from 'drugName' and 'condition' and convert to lowercase for standardization

UPDATE dbo.drugs\_dataset

SET condition = LOWER(TRIM(condition)),

drugName = LOWER(TRIM(drugName)); -- Standardize text and ensure uniform formatting

-- Remove any duplicate rows based on the 'id' column

WITH CTE AS (

SELECT \*,

ROW\_NUMBER() OVER (PARTITION BY id ORDER BY id) AS rn

FROM dbo.drugs\_dataset

)

DELETE FROM CTE WHERE rn > 1; -- Keep only the first occurrence of each ID

-- Identify any outliers in 'rating' where the value is not between 1 and 10

SELECT \*

FROM dbo.drugs\_dataset

WHERE rating < 1 OR rating > 10; -- Ensure ratings are within expected range

---- FEATURE ENGINEERING ----

-- Add a new column 'word\_count' to count the number of words in each review

ALTER TABLE dbo.drugs\_dataset ADD word\_count INT;

-- Populate 'word\_count' with the number of words in each review

UPDATE dbo.drugs\_dataset

SET word\_count = LEN(review) - LEN(REPLACE(review, ' ', '')) + 1; -- Calculate word count

-- Split the 'date' column into separate 'year', 'month', and 'day' columns for easier time-based analysis

ALTER TABLE dbo.drugs\_dataset ADD year INT;

ALTER TABLE dbo.drugs\_dataset ADD month INT;

ALTER TABLE dbo.drugs\_dataset ADD day INT;

-- Populate the 'year', 'month', and 'day' columns with data extracted from the 'date' column

UPDATE dbo.drugs\_dataset

SET year = YEAR(date),

month = MONTH(date),

day = DAY(date); -- Split the date into separate components

select \* from dbo.drugs\_dataset

-- Calculate mean and standard deviation for 'usefulCount' to identify outliers

WITH Stats AS (

SELECT

AVG(usefulCount) as MeanUsefulCount,

STDEV(usefulCount) as StdDevUsefulCount

FROM dbo.drugs\_dataset

),

Outliers AS (

SELECT \*,

(usefulCount - Stats.MeanUsefulCount) / Stats.StdDevUsefulCount as ZScore

FROM dbo.drugs\_dataset, Stats

)

SELECT \*

FROM Outliers

WHERE ABS(ZScore) > 3; -- Identify reviews where 'usefulCount' is an outlier

-- Delete outliers based on the calculated Z-Score

WITH Stats AS (

SELECT

AVG(usefulCount) as MeanUsefulCount,

STDEV(usefulCount) as StdDevUsefulCount

FROM dbo.drugs\_dataset

),

Outliers AS (

SELECT \*,

(usefulCount - Stats.MeanUsefulCount) / Stats.StdDevUsefulCount as ZScore

FROM dbo.drugs\_dataset, Stats

)

DELETE FROM dbo.drugs\_dataset

WHERE ID IN (SELECT ID FROM Outliers WHERE ABS(ZScore) > 3); -- Remove outliers ----(3929 rows affected)

select COUNT(\*) from dbo.drugs\_dataset ---209940 rows

-- Check for any leading or trailing spaces in 'drugName' and 'condition'

SELECT \*

FROM dbo.drugs\_dataset

WHERE LTRIM(RTRIM(drugName)) <> drugName

OR LTRIM(RTRIM(condition)) <> condition; -- Identify records with unintended spaces

-- Add a 'sentiment' column to classify reviews as positive (1), negative (-1), or neutral (0)

ALTER TABLE dbo.drugs\_dataset ADD sentiment INT;

-- Populate the 'sentiment' column based on keywords in the review

UPDATE dbo.drugs\_dataset

SET sentiment = CASE

WHEN LOWER(review) LIKE '%good%' OR LOWER(review) LIKE '%great%' OR LOWER(review) LIKE '%excellent%' THEN 1 -- Positive sentiment

WHEN LOWER(review) LIKE '%bad%' OR LOWER(review) LIKE '%terrible%' OR LOWER(review) LIKE '%poor%' THEN -1 -- Negative sentiment

ELSE 0 -- Neutral sentiment

END;

-- Check the table after sentiment analysis

SELECT \* FROM dbo.drugs\_dataset;

---- BINARIZATION ----

-- Add a binary column 'rating\_binary' to classify ratings as high (1) or low (0)

ALTER TABLE dbo.drugs\_dataset ADD rating\_binary VARCHAR(10);

-- Populate 'rating\_binary' based on the 'rating' column (1 if rating >= 7, else 0)

UPDATE dbo.drugs\_dataset

SET rating\_binary = CASE

WHEN rating >= 7 THEN '1' -- High rating

ELSE '0' -- Low rating

END;

-- Final check of the processed data

SELECT \* FROM dbo.drugs\_dataset;

select COUNT(\*) from dbo.drugs\_dataset

select top 1500 \* from dbo.drugs\_dataset